IPA Dokumentation 2014

|  |  |  |  |
| --- | --- | --- | --- |
| Kunde | 4teamworkAG | | |
| Projektname | Automatische Installation von Demo-Webplattformen | | |
| Projektnummer | 1 | | |
| Autor | Michel Weingart | | |
| Ausgabedatum | XXXXX | | |
| Version | V 0.1 | | |
| Status | In Arbeit | In Prüfung | Genehmigt, zur Nutzung |
|  |  |  |  |

| Beteiligter Personenkreis | |
| --- | --- |
| Benutzer, Anwender | Michel Weingart, IT Support |
| Prüfung | Pascal Habegger, Fachvorgesetzter Manuel J. Schaffner, Hauptexperte Verena Meinung, 2. Expertin |
| Genehmigung | Yvan Kolly, Validexpert |
| zur Information, Kenntnis | Josef Müller, Berufsbildner |

Es handelt sich hierbei um eine Beispielvorlage, welche unbedingt dem entsprechenden Projekt angepasst werden muss.

**Die Elemente des Teil 1 sind verbindlich!**

Nicht alle Elemente im Teil 2 müssen genutzt werden.

Die Hilfstexte sind nur sichtbar, wenn das Pi Zeichen

(π = Absatzmarken / Formatierungssymbole) eingeschaltet ist!

Es wird auf die weibliche Form verzichtet, aus einfachheitsgründen.

(Diese Textbox kann gelöscht werden)

Dokumentinformationen

Änderungskontrolle, Prüfung, Genehmigung

In der Änderungskontrolle werden nur Punkte aufgelistet, welche vollständig in der Dokumentation enthalten ist, weshalb der Arbeitsplan nicht 100% mit der Änderungskontrolle übereinstimmen wird.

|  | | | |
| --- | --- | --- | --- |
| Version | Datum | Name | Beschreibung |
| Vorlage | 2014-05-05 |  | IPA\_Dokumentation\_2014-final |
| V0.1 | 2014-05.05 | Michel Weingart | * Dokumenteröffnung * Anpassung der Struktur / Standardeintragungen * Anpassung der Struktur * Erarbeitung Zeitplan * Erfassung Aufgabenstellung * Erfassung Vorkenntnisse * Erfassung Vorarbeiten * Erfassung Firmenstandards * Erstellung Projektantrag * Definition Organisation des Projekts |
| V0.2 | 2014-05.06 | Michel Weingart | * Erarbeitung Studio Ist / Soll * Erstellung der Vorgehensziele * Erstellung der Systemziele * Definieren der Anforderungen * Erarbeitung des Variantenentscheids * Erarbeitung der Informationssicherheit und des Datenschutzes * Erarbeitung Lösungssuche |

Referenzierte Dokumente

Verweis auf andere Dokumente wie Projektdoku ...

| Ref. | Beschreibung |
| --- | --- |
| Filename | Titel des Dokuments |
|  |  |

Verwendete Abkürzungen

Nur Abkürzungen, keine Begriffserklärungen. Diese gehören in ein Glossar. Alphabetisch sortiert.

| Abkürzung | Bedeutung |
| --- | --- |
| IPA | Individuelle praktische Arbeit |
| OdA | Organisation der Arbeitwelt |
| QV | Qualifikationsverfahren |
| VZ | Verzeichnis |

Inhaltsverzeichnis

[Dokumentinformationen 2](#_Toc373508046)

[Inhaltsverzeichnis 4](#_Toc373508047)

[Abbildungsverzeichnis 6](#_Toc373508048)

[Tabellenverzeichnis 6](#_Toc373508049)

[Management Summary 7](#_Toc373508050)

[Teil 1: Ablauf und Umfeld 8](#_Toc373508051)

[1. Aufgabenstellung 9](#_Toc373508052)

[1.1 Ausgangslage 9](#_Toc373508053)

[1.2 Auftragsformulierung 9](#_Toc373508054)

[1.3 Mittel und Methoden 10](#_Toc373508055)

[1.4 Projektmanagementplan 10](#_Toc373508056)

[1.5 Projektrollen 11](#_Toc373508057)

[2. Vorkenntnisse Error! Bookmark not defined.](#_Toc373508058)

[3. Vorarbeiten 13](#_Toc373508059)

[4. Firmenstandards 14](#_Toc373508060)

[5. Organisation der IPA 16](#_Toc373508061)

[5.1 Datensicherung der IPA 17](#_Toc373508062)

[6. Zeitplan 18](#_Toc373508063)

[7. Arbeitsjournal 19](#_Toc373508064)

[7.1 Erster Tag: Montag, xx.yy.20xx 20](#_Toc373508065)

[7.2 Zweiter Tag: Montag, xx.yy.20xx 22](#_Toc373508066)

[7.3 Dritter Tag: Montag, xx.yy.20xx 23](#_Toc373508067)

[7.4 Vierter Tag: Montag, xx.yy.20xx 24](#_Toc373508068)

[7.5 Fünfter Tag: Montag, xx.yy.20xx 25](#_Toc373508069)

[7.6 Sechter Tag: Montag, xx.yy.20xx 26](#_Toc373508070)

[7.7 Siebter Tag: Montag, xx.yy.20xx 27](#_Toc373508071)

[7.8 Achter Tag: Montag, xx.yy.20xx 28](#_Toc373508072)

[7.9 Neunter Tag: Montag, xx.yy.20xx 29](#_Toc373508073)

[7.10 Zehnter Tag: Montag, xx.yy.20xx 30](#_Toc373508074)

[7.11 Arbeitszeit total 31](#_Toc373508075)

[8. Abschlussbericht 34](#_Toc373508076)

[8.1 Vergleich Ist/Soll 34](#_Toc373508077)

[8.2 Mittelbedarf 34](#_Toc373508078)

[8.3 Realisierungsbericht 34](#_Toc373508079)

[8.4 Testbericht 34](#_Toc373508080)

[8.5 Fazit zum Projekt 34](#_Toc373508081)

[8.6 Persönliches Fazit 34](#_Toc373508082)

[9. Unterschriften Teil 1 35](#_Toc373508083)

[Teil 2: Projektdokumentation 36](#_Toc373508084)

[10. Initialisierung 37](#_Toc373508085)

[10.1 Studie Ist Zustand / Soll-Zustand 37](#_Toc373508086)

[10.2 Vorgehensziele 38](#_Toc373508087)

[10.3 Systemziele 38](#_Toc373508088)

[10.4 Anforderungen 40](#_Toc373508089)

[10.4.1 Funktionale Anforderungen 40](#_Toc373508090)

[10.4.2 Nicht funktionale Anforderungen 40](#_Toc373508091)

[10.5 Risikoanalyse 41](#_Toc373508092)

[10.6 Risikograph **Error! Bookmark not defined.**](#_Toc373508093)

[10.7 Varianten 42](#_Toc373508094)

[10.7.1 Variante 1 42](#_Toc373508095)

[10.7.2 Variante 2 42](#_Toc373508096)

[10.7.3 Variante 3 **Error! Bookmark not defined.**](#_Toc373508097)

[10.8 Variantenentscheid 43](#_Toc373508098)

[10.8.1 Begründung 43](#_Toc373508099)

[10.9 Informationssicherheit und Datenschutz (ISDS) 44](#_Toc373508100)

[10.10 Lösungen suchen 44](#_Toc373508101)

[11. Konzept 47](#_Toc373508102)

[11.1 Konzept entwickeln 47](#_Toc373508103)

[11.2 Systemdesign erstellen 47](#_Toc373508104)

[11.3 Schutzmassnahmen erarbeiten 47](#_Toc373508105)

[11.4 Testkonzept. 48](#_Toc373508106)

[12. Realisierung 49](#_Toc373508107)

[12.1 System erstellen 49](#_Toc373508108)

[12.2 Testprotokoll 49](#_Toc373508109)

[12.3 Einführung vorbereiten 50](#_Toc373508110)

[12.4 Schutzmassnahmen umsetzen 50](#_Toc373508111)

[13. Einführung 51](#_Toc373508112)

[13.1 System einführen 51](#_Toc373508113)

[14. Quellenverzeichnis 52](#_Toc373508114)

[15. Glossar 53](#_Toc373508115)

[16. Unterschriften für Abnahme 54](#_Toc373508116)

[17. Anhang 55](#_Toc373508117)

Abbildungsverzeichnis

Hinweis: An dieser Stelle kann ein Abbildungsverzeichnis eingefügt werden.

[Abbildung 1: Projektorganisation 10](#_Toc373508118)

[Abbildung 2: Hermes 5 IPA 36](#_Toc373508119)

Tabellenverzeichnis

Hinweis: An dieser Stelle kann ein Tabellenverzeichnis eingefügt werden.

[Tabelle 1: Projektrollen 11](#_Toc376847981)

[Tabelle 2: Risikoanalyse Tabelle **Error! Bookmark not defined.**](#_Toc376847982)

[Tabelle 3: Risikograph **Error! Bookmark not defined.**](#_Toc376847983)

[Tabelle 4: Variantenentscheid 43](#_Toc376847984)

[Tabelle 5: Testprotokoll 49](#_Toc376847985)

Management Summary

Zusammenfassung der IPA-Resultate; max. einer A4 Seite (Management Summary).

max. zwei A4 Seiten Abschlussbericht (zweitletzte Seite Teil 1).

Ausgangssituation

Umsetzung

Ergebnis

Teil 1: Ablauf und Umfeld

IPA Projektname: Automatische Installation von Demo-Webplattformen

Autor: Michel Weingart



# Aufgabenstellung

Für mehrere webbasierte Standardlösungen soll regelmässig (e.g. wöchentlich) und automatisiert öffentliche Demoinstallationen mit Beispielinhalten im Internet verfügbar gemacht werden.

## Ausgangslage

4teamwork entwickelt mehrere webbasierte Standardlösungen: teamraum Collab, teamraum Intranet, teamraum Web. Diese Lösungen basieren auf dem Content Management System Plone und werden in einem geordneten Releaseprozess laufend erneuert und erweitert. Neue Funktionen werden verfügbar gemacht, von den Benutzern gemeldete Fehler und Unzulänglichkeiten werden korrigiert. Es existiert eine Demoplattform demo.teamraum.ch, über die sich interessierte Personen einloggen und die Lösung unverbindlich testen können. Die Demoplattform wird von 4teamwork nur unregelmässig aktualisiert und ist daher meistens nicht aktuell. Sie enthält kaum Beispielinhalte.  
  
Entwicklungs- und Testinstallationen der Standardlösungen werden von den Entwicklern von 4teamwork zwar immer wieder neu erstellt, allerdings sind diese nur lokal verfügbar, und nicht von aussen zugänglich. Diese Installationen dienen vorwiegend für automatisierte Softwaretests, ohne Benutzerinteraktionen. Aus diesen Gründen werden neue Releases von teamraum unter Umständen zu wenig von Menschen getestet (Human Testing), bevor sie freigegeben werden.

## Auftragsformulierung

Es werden die folgenden Resultate erwartet:

1) Es wird ein Konzept entwickelt, wie die Plattformen mit Beispielinhalten und -benutzern automatisiert installiert und betrieben werden können. Zwei Hauptvarianten sind dabei denkbar und müssen nach verschiedenen Kriterien bewertet werden:  
1) Bereitstellen eines VMWare-Image mit einem vorbereiteten Release einer Webanwendung inkl. Beispielinhalten und -benutzer. Das VMWare-Image kann dann regelmässig neu geladen werden.  
2) Auf einer bestehenden Linux-Umgebung werden per Buildout (d.h. geskriptet) automatisiert neue Webinstallationen installiert und gestartet.  
Die höher bewertete Variante soll umgesetzt werden (siehe 2)).  
  
2) Für eine der oben genannten Standardlösungen von teamraum soll eine automatisierte Demoplattform im Internet installiert werden, so dass interessierte Personen die Lösungen selbständig antesten können. Die Demoplattform verwendet immer den neuesten, freigegebenen Release und enthalten Anwendungsfall bezogene Beispielinhalte. Der Entscheid, welche Standardlösung automatisiert installiert werden soll, wird dem Lernenden überlassen. Für die gewählte Standardlösung müssen dann anwendungsbezogene Beispielinhalte und -benutzer definiert werden. Diese Inhalte werden dann bei jeder Neuinstallation der Demoumgebung automatisch hinzugefügt.  
  
3) Der Prozess, der verwendet wird, um die Demoplattformen von neuen Releases von teamraum automatisiert zu installieren, soll auch für künftige Entwicklungs- und Testreleases eingesetzt werden können. So können sog. Release Candidates von teamraum von einem kleineren Benutzerkreis über das Web manuell getestet werden. Damit lässt sich die Qualität eines teamraum Release weiter steigern. Der geforderte Prozess muss technisch gut dokumentiert sein, damit auch andere Personen in Zukunft derartige Testsysteme bereitstellen können.

## Mittel und Methoden

* Programmierung und Dokumentation werden auf einem Mac-Book durchgeführt.
* Die Dokumentation wird ein Word2010 verfasst.
* Als Programmiersprachen kommen (in geringem Umfang) Python und Bash-Skripte zum Einsatz.
* Als Programmierungsumgebung wird der Texteditor Sublime Text2 verwendet
* Gute Kenntnisse von X-HTML und XML sind erforderlich.
* Anwendungswissen von Plone und teamraum ist erforderlich.
* Die Demoinstallationen können lokal auf Mac Notebook oder auf einem Linux-Server von 4teamwork/in einer Cloud-Umgebung (Variantenentscheid) gemacht werden.

## Projektmanagementplan

**Qualität- & Sicherheitsmanager**

Valid-Experte: Yvan Kolly   
Hauptexperte: Manuel J. Schaffner  
2. Expertin: Verena Meinung

**Auftraggeber:**

4teamwork AG

**Projektleiter:**

Michel Weingart

**Fachvorgesetzter**

Dr. Pascal Habegger

**Fachspezialist Technik**

Mathias Leimgruber

**Testing**

Dr. Thomas Buchberger

**Projektausschuss**

Dr. Pascal Habegger

Abbildung : Projektorganisation

## Projektrollen

|  |  |
| --- | --- |
| Auftraggeber: | 4teamwork AG |
| Projektausschuss | Stellt das oberste beschlussfassende Gremium einer Projektorganisation dar. |
| Valid-Experte | Yvan Kolly |
| Hauptexperte | Manuel J. Schaffner |
| 2. Expertin | Verena Meinung |
| Projektleiter | Michel Weingart |
| Fachvorgesetzter | Dr. Pascal Habegger |
| Stellvertretender Fachvorgesetzter | Mathias Leimgruber |
| Fachspezialist Technik | Mathias Leimgruber |
| Testperson | Dr. Thomas Buchberger |

Tabelle : Projektrollen

# Vorkenntnisse

Eine Übersicht über die Vorkenntnisse, welche für die Durchführung dieses Projekts relevant sind.

|  |  |  |
| --- | --- | --- |
| Anwendung | Einsatzbereich | Dauer |
| Webanwendung Teamraum | Schulung und Supportarbeiten | 2 Jahre |
| Browser (Chrome, Firefox, Safari, Internet Explorer) | Alltag, Support | 2 Jahre |
| Serveradministration Linux Grundwissen | Supportarbeiten | 1Jahr |
| Entwicklung mit Programmiersprache Python | - | - |
| Plone Grundwissen | Supportarbeiten | 1Jahr |
| Git | Kleinere Entwicklungen | 0.5 Jahr |

# Vorarbeiten

Es wurden folgende Arbeiten durchgeführt:

* Diverse Vorbesprechungen mit dem Fachvorgesetzten zur Definition des IPA-Themas
* Einarbeitung in Plone, wie Webinstallation technisch mit Inhalten vorbefüllt werden können
* Besprechung mit Entwickler betreffend Programmiersprache Python
* Programmierung von Python mit Hilfe Entwickler
* Einrichtung eines Arbeitsplatzes für die IPA
* Erstellen eines Kanban-Plakats
* Dokumente von durchgeführten IPAs durchgelesen

# Firmenstandards

Da bei der Firma 4teamwork AG keine Standards betreffend Vorlagen existieren, benutze ich die Dokumentvorlage welche von Pkorg zur Verfügung gestellt ist.

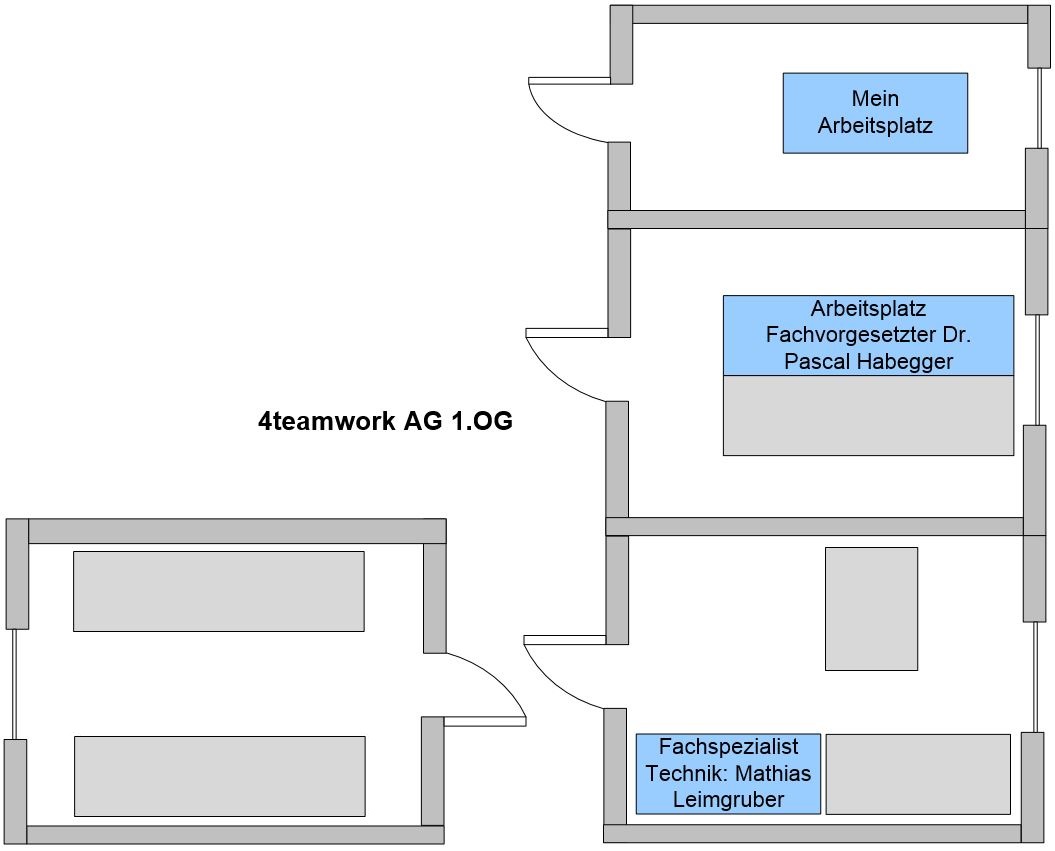
# Meilensteine

In diesem Projekt sind folgende Meilensteine definiert:

|  |  |  |
| --- | --- | --- |
| Datum | Beschreibung | Erreicht |
| 05.05.2014 | Start der IPA | Ja |
| 06.05.2014 | Phase Initialisierung ist abgeschlossen und Phase Konzept freigegeben | Ja |
| 19.05.2014 | Phase Konzept ist abgeschlossen und Phase Realisierung freigegeben |  |
| 22.05.2014 | Phase Realisierung ist abgeschlossen und Phase Einführung freigegeben |  |
| 26.05.2014 | Phase Einführung ist abgeschlossen und Dokument abgeschlossen. |  |

# Organisation der IPA

Um die IPA ungestört durchführen zu können habe ich meinen Arbeitsplatz, mit dem nötigen Zubehör, vom Supportzimmer in ein meist ungenutztes Zimmer verlegt.





Zusätzlich zum Wechsel des Arbeitsplatzes, habe ich mich entschlossen das Vorgehensmodell „Kanban“ zu benutzen. Dies wurde bei der Firma 4teamwork vor ca. einem halben Jahr eingeführt und hat sich sowohl bei den Entwicklern als auch bei den Projektmanagern bewährt. Dies dient mir zur vereinfachten Aufgabenteilung und schnelleren Erkennung von Problemen. 

## Datensicherung der IPA

Die Datensicherung und Dokumentablage wird mittels privaten Git Repositorys umgesetzt. Somit ist eine automatische Versionierung gewährleistet, wird ein Datenverlust vermieden und der Datenschutz eingehalten.

|  |  |  |
| --- | --- | --- |
| Name Repository | Pfad | Beschreibung |
| IPA\_michel.weingart | https://github.com/4teamwork/IPA\_michel.weingart | Ablage von:  - Dokumenten  - Quellcode  - Bilder  - Grafiken |

# Zeitplan

Planung und Umsetzung in Form von A3 Querseiten (entsprechend falten)

Individueller Zeitplan (nach Projektmethode).¨

# Arbeitsjournal

Im Arbeitsjournal sind die wichtigsten Informationen chronologisch gesammelt. Täglich werden das Vorgehen, fremde Hilfestellungen und besondere Vorkommnisse dokumentiert.

Die Festlegungen dieses Dokuments gelten im Projekt.

Gemäss Art. 5 Absatz 2 der Wegleitung über die individuelle praktische Arbeit (IPA) an Lehrabschlussprüfungen des BBT vom 27. August 2001 gilt:

*„Die zu prüfende Person führt ein Arbeitsjournal. Sie dokumentiert darin täglich das Vorgehen, den Stand der Prüfungsarbeit, sämtliche fremde Hilfestellungen und besondere Vorkommnisse wie z.B. Änderungen der Aufgabenstellung, Arbeitsunterbrüche, organisatorische Probleme, Abweichungen von der Soll-Planung.“*

Das Arbeitsjournal zur IPA ist zwingend zu führen und den Experten und Fachvorgesetzten vorzulegen. Das Arbeitsjournal ist täglich sinngemäss und korrekt auszufüllen.

Das Arbeitsjournal dient der Nachvollziehbarkeit der von den Lernenden ausgeführten Arbeiten und wird als Teil der IPA in die Bewertung mit einbezogen.

**Beteiligte Personen:**

MW -> Michel Weingart

MS -> Manuel Schaffner

VM -> Verena Meinung

PH -> Pascal Habegger

ML -> Mathias Leimgruber

TB -> Thomas Buchberger

Das Arbeitsjournal ist den Experten beim Besuch vorzulegen. Es ist sinnvoll, korrekt (mit den echten Zeitangaben) jeden Tag im Stundenraster zu führen.

Arbeitsprotokoll (Journal)

Nachvollziehbare tägliche Notizen (mit Datum) zu den ausgeführten Arbeiten

Erreichte Ziele / Anforderungen

Aufgetretene Probleme

Erfolgreiche oder erfolglose Tests

Beanspruchte Hilfestellung (wer, was)

Nacht- und Wochenendarbeit

Vergleich mit Zeitplan

Kritische Hinterfragung des Tages

Kritischer Rückblick (Reflexion) was war gut? was habe ich gelernt? was könnte ich besser machen? auf was bin ich stolz?

## Erster Tag: Montag, 05.05.2014

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Person | Aufwand  geplant (Std) | Aufwand  effektiv (Std) |
| *Kickoff-Meeting*  Das Kickoff-Meeting wurde mit dem stellvertretenden Fachvorgesetzten Mathias Leimgruber durchgeführt. Das Projekt kann beginnen. | MW/ML | 0.5 | 0.5 |
| *Git Repository*  Ein Git Repository wurde unter „ [https://github.com/4teamwork“ unter dem Namen „IPA\_michel.weingart.git](%20https://github.com/4teamwork)“ erstellt um die Dokumentversionen und sonstige Daten hochladen zu können. | MW | 0.5 | 0.5 |
| *Dokumentation eröffnen*  Die Dokumentation wurde eröffnet. | MW | 0.5 | 0.5 |
| *Standardeintragungen in der Dokumentation*  Ich habe die ersten Eintragungen in der Dokumentation vorgenommen wie z.B Header / Footer... | MW | 0.5 | 0.5 |
| *Zeitplan erstellen*  Ein Zeitplan wurde erstellt und die Soll-Zeiten eingetragen. | MW | 2.0 | 2.5 |
| *Aufgabenstellung erfassen*  Die Aufgabenstellung zum Projekt wurde erfasst. | MW | 1.0 | 1.5 |
| *Vorkenntnisse / Vorarbeiten / Firmenstandards / Organisaiton des Projekts erfassen*  Elemente der Dokumente wurden eingepflegt. | MW | 1.0 | 1.0 |
| *Studie Ist / Soll*  Kurze Reflektion betreffend des Ist und Soll -Zustandes wurde gemacht. | MW | 0.5 | 0.5 |
| *Eintrag Arbeitsjournal*  Das Arbeitsjournal wurde nachgeführt. | MW | 1.0 | 1.0 |
| Total: |  | 7.5 | 8.5 |
| Tages Ablauf |  |  |  |
| Das Projekt konnte ich nach dem Kickoff-Meeting erfolgreich starten. Nach kleineren Problemen betreffend dem Git Repository, habe ich die Dokumentation eröffnet.  Einige Probleme bereiteten mir auch das Erstellen des Zeitplans. Dieser beanspruchte mehr Zeit als eingeplant, da ich an ein Excel auf Windows-Systemen gewöhnt bin und selten auf dem MacBook Excel verwende, bei welchem die Software anders aufgebaut ist. Das schätzen des Zeitaufwands für die verschiedenen Phasen habe ich mir einfacher vorgestellt, hoffe aber dass die Schätzungen realistisch sind. Dafür gestaltete sich das Erfassen der ersten Einträge in der Dokumentation leichter als erwartet, was wiederum einen kleinen Zeitgewinn erbrachte. | | | |
| Hilfestellungen |  |  |  |
| Ich habe anfangs lokal eine Ordnerstruktur im neuen Git Repository erstellt. Mit dem Befehl „git status“ sind keine Änderungen erkannt worden. Habe bei einem Entwickler nachgefragt, ob Ordner in einem Git Repository nicht als neue Eintragungen erkannt werden. Habe erfahren, dass diese nicht erkannt werden, sondern nur Daten. | | | |
| Reflexion |  |  |  |
| Die heutigen Ziele habe ich grundsätzlich erreicht und der Start verlief sehr gut. Ich konnte ohne grosse Probleme mit der Dokumentation beginnen.  Das Einarbeiten in das Programm Excel auf dem Mac hat Zeit benötigt, welche ich nicht zur Verfügung habe. Die bestehenden Unterschiede von Software unter Windows und unter Mac sollte ich in Zukunft besser berücksichtigen. | | | |
| Nächste Schritte |  |  |  |
| Aufgaben welche ich für morgen geplant habe:  - Fertigstellung der Voranalyse  - Variantenentscheid durchführen  - Risikoanalyse durchführen  - Freigabe Konzept einholen | | | |

## Zweiter Tag: Dienstag, 06.05.2014

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Person | Aufwand  geplant (Std) | Aufwand  effektiv (Std) |
| *Studie Ist / Soll*  Der aktuelle und zu erwartende Zustand wurde aufgezeigt und der Prozess abgebildet. | MW | 1.5 | 1.0 |
| *Vorgehensziele / Systemziele / Anforderungen*  Elemente der Dokumente wurden eingepflegt. | MW | 1.5 | 1.5 |
| *Variantenentscheid*  Die bessere Variante konnte anhand des Variantenentscheids eruiert werden. | MW | 1.5 | 1.5 |
| *Informationssicherheit und Datenschutz /*  *Lösungen suchen*  Elemente der Dokumente wurden eingepflegt. | MW | 1.5 | 1.5 |
| *Freigabe Konzept*  Die Phase Konzept wurde freigegeben. | MW/PH | 0.5 | 0.5 |
| *Eintrag Arbeitsjournal*  Das Arbeitsjournal wurde nachgeführt. | MW | 1.0 | 1.0 |
| Total: |  | 7.5 | 7.0 |
| Tages Ablauf |  |  |  |
| Der Ist / Soll Vergleich konnte ich gut abschliessen und grafisch darstellen.  Die bessere Variante hat sich deutlicher herauskristallisiert als ich dies angedacht habe, was den Entscheid für die Variante per Script sehr vereinfachte. Auch war die Lösungssuche mit der Methode „Brainstorming“ sehr aufschlussreich und ich konnte die wichtigsten Kriterien daraus gewinnen. Hingegen bereitete mir das Einfüllen der Vorgehensziele / Systemziele und Anforderungen Mühe, da die Anforderungen dazu für mich nicht klar ersichtlich waren. | | | |
| Hilfestellungen |  |  |  |
| Für das Einfüllen der Vorgehensziele / Systemziele und Anforderungen konnte ich mir einige Tipps vom Fachspezialist Technik Mathias Leimgruber holen. Auch für die Auswahl der Kriterien für den Variantenentscheid habe ich seine Hilfe beansprucht. | | | |
| Meinem Hauptexperten habe ich eine Fragestellung betreffend Änderungen des Zeitplans zugestellt. Mit war nicht klar, wie eine Änderung erfasst werden soll und ob der Zeitplan überhaupt angepasst werden darf. Falls der Hauptexperte die Frage nicht beantworten kann, werde ich diese beim ersten Expertenbesuch stellen. | | | |
| Reflexion |  |  |  |
| Trotz des Zeitgewinnes bei der Studie zum Ist / Soll Vergleich habe ich dort folgenden Punkt zur Verbesserung: Die Aufteilung der Arbeiten über mehrere Tage habe ich nicht als förderlich empfunden und werde dies in Zukunft vermeiden. Die Studie habe ich am Montag angefangen und am Dienstag beendet. Der Unterbruch verlangte, dass ich mich zuerst wieder in das Thema einarbeiten musste. | | | |
| Nächste Schritte |  |  |  |
| Ich werde mich auf den Erstbesuch des Hauptexperten vorbereiten und Fragen bezüglich der IPA aufschreiben um diese ihm stellen zu können.  Weitere Aufgaben welche ich für morgen geplant habe:   * Situationsanalyse * Systemanforderungen ermitteln * Zugriffskonzept erstellen * Erarbeitung des Betriebskonzepts | | | |

## Dritter Tag: Mittwoch, 07.05.2014

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Person | Aufwand  geplant (Std) | Aufwand  effektiv (Std) |
| *Überschrift*  Beschreibung |  |  |  |
| *Überschrift*  Beschreibung |  |  |  |
| Total: |  |  |  |
| Tages Ablauf |  |  |  |
| Beschreibung von Erfolgen und Misserfolgen (Problemen) | | | |
| Hilfestellungen |  |  |  |
| Heute Morgen habe ich noch die Rückmeldung des Experten betreffend der gestrigen Fragestellung (Anpassung Zeitplan) erhalten. Er hat mich darauf hingewiesen, dass die Soll-Zeiten angepasst werden können, hingegen die Ist-Zeiten so belassen werden sollen. | | | |
|  | | | |
| Reflexion |  |  |  |
| Gedanken über das Vorgehen  Was ist gut was würde ich besser machen | | | |
| Nächste Schritte |  |  |  |
| Wie weiter, nächste Schritte... | | | |

## Vierter Tag: Donnerstag, 08.05.2014

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Person | Aufwand  geplant (Std) | Aufwand  effektiv (Std) |
| *Überschrift*  Beschreibung |  |  |  |
| *Überschrift*  Beschreibung |  |  |  |
| Total: |  |  |  |
| Tages Ablauf |  |  |  |
| Beschreibung von Erfolgen und Misserfolgen (Problemen) | | | |
| Hilfestellungen |  |  |  |
| Wer hat in welcher Form Hilfestellung erbracht | | | |
|  | | | |
| Reflexion |  |  |  |
| Gedanken über das Vorgehen  Was ist gut was würde ich besser machen | | | |
| Nächste Schritte |  |  |  |
| Wie weiter, nächste Schritte... | | | |

## Fünfter Tag: Montag, xx.yy.20xx

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Person | Aufwand  geplant (Std) | Aufwand  effektiv (Std) |
| *Überschrift*  Beschreibung |  |  |  |
| *Überschrift*  Beschreibung |  |  |  |
| Total: |  |  |  |
| Tages Ablauf |  |  |  |
| Beschreibung von Erfolgen und Misserfolgen (Problemen) | | | |
| Hilfestellungen |  |  |  |
| Wer hat in welcher Form Hilfestellung erbracht | | | |
|  | | | |
| Reflexion |  |  |  |
| Gedanken über das Vorgehen  Was ist gut was würde ich besser machen | | | |
| Nächste Schritte |  |  |  |
| Wie weiter, nächste Schritte... | | | |

## Sechter Tag: Montag, xx.yy.20xx

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Person | Aufwand  geplant (Std) | Aufwand  effektiv (Std) |
| *Überschrift*  Beschreibung |  |  |  |
| *Überschrift*  Beschreibung |  |  |  |
| Total: |  |  |  |
| Tages Ablauf |  |  |  |
| Beschreibung von Erfolgen und Misserfolgen (Problemen) | | | |
| Hilfestellungen |  |  |  |
| Wer hat in welcher Form Hilfestellung erbracht | | | |
|  | | | |
| Reflexion |  |  |  |
| Gedanken über das Vorgehen  Was ist gut was würde ich besser machen | | | |
| Nächste Schritte |  |  |  |
| Wie weiter, nächste Schritte... | | | |

## Siebter Tag: Montag, xx.yy.20xx

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Person | Aufwand  geplant (Std) | Aufwand  effektiv (Std) |
| *Überschrift*  Beschreibung |  |  |  |
| *Überschrift*  Beschreibung |  |  |  |
| Total: |  |  |  |
| Tages Ablauf |  |  |  |
| Beschreibung von Erfolgen und Misserfolgen (Problemen) | | | |
| Hilfestellungen |  |  |  |
| Wer hat in welcher Form Hilfestellung erbracht | | | |
|  | | | |
| Reflexion |  |  |  |
| Gedanken über das Vorgehen  Was ist gut was würde ich besser machen | | | |
| Nächste Schritte |  |  |  |
| Wie weiter, nächste Schritte... | | | |

## Achter Tag: Montag, xx.yy.20xx

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Person | Aufwand  geplant (Std) | Aufwand  effektiv (Std) |
| *Überschrift*  Beschreibung |  |  |  |
| *Überschrift*  Beschreibung |  |  |  |
| Total: |  |  |  |
| Tages Ablauf |  |  |  |
| Beschreibung von Erfolgen und Misserfolgen (Problemen) | | | |
| Hilfestellungen |  |  |  |
| Wer hat in welcher Form Hilfestellung erbracht | | | |
|  | | | |
| Reflexion |  |  |  |
| Gedanken über das Vorgehen  Was ist gut was würde ich besser machen | | | |
| Nächste Schritte |  |  |  |
| Wie weiter, nächste Schritte... | | | |

## Neunter Tag: Montag, xx.yy.20xx

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Person | Aufwand  geplant (Std) | Aufwand  effektiv (Std) |
| *Überschrift*  Beschreibung |  |  |  |
| *Überschrift*  Beschreibung |  |  |  |
| Total: |  |  |  |
| Tages Ablauf |  |  |  |
| Beschreibung von Erfolgen und Misserfolgen (Problemen) | | | |
| Hilfestellungen |  |  |  |
| Wer hat in welcher Form Hilfestellung erbracht | | | |
|  | | | |
| Reflexion |  |  |  |
| Gedanken über das Vorgehen  Was ist gut was würde ich besser machen | | | |
| Nächste Schritte |  |  |  |
| Wie weiter, nächste Schritte... | | | |

## Zehnter Tag: Montag, xx.yy.20xx

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Person | Aufwand  geplant (Std) | Aufwand  effektiv (Std) |
| *Überschrift*  Beschreibung |  |  |  |
| *Überschrift*  Beschreibung |  |  |  |
| Total: |  |  |  |
| Tages Ablauf |  |  |  |
| Beschreibung von Erfolgen und Misserfolgen (Problemen) | | | |
| Hilfestellungen |  |  |  |
| Wer hat in welcher Form Hilfestellung erbracht | | | |
|  | | | |
| Reflexion |  |  |  |
| Gedanken über das Vorgehen  Was ist gut was würde ich besser machen | | | |
| Nächste Schritte |  |  |  |
| Wie weiter, nächste Schritte... | | | |

Weitere Arbeitstage wie Wochenendarbeit hier auch vermerken.

## Arbeitszeit total

|  |  |  |  |
| --- | --- | --- | --- |
| Totaler Zeitaufwand | Person | Aufwand  geplant (Std) | Aufwand  effektiv (Std) |
|  |  |  |  |
| Reflexion |  |  |  |
| Kurze Begründung falls unterschritten oder überzogen (Richtwert 80 Std +/-10%) | | | |

# Projektjournal

Im Projektjournal sind Informationen chronologisch gesammelt, welche im Verlauf

der Arbeit eine Rolle spielen. Besprechungsprotokolle mit Entscheiden und Abmachungen

sind besonders wichtig. Das Projektjournal wird laufend abgefüllt.

|  |  |
| --- | --- |
| Datum: | 05.05.2014 |
| Typ: | Meilenstein |
| Beteiligte Personen: | Stellvertretender Fachvorgesetzter, Mathias Leimgruber |
| Themen: | Start der IPA |
| Ergebnis: | IPA wurde nach dem Kickoff-Meeting gestartet |

|  |  |
| --- | --- |
| Datum: | 05.05.2014 |
| Typ: | Telefonbesprechung |
| Beteiligte Personen: | * Hauptexperte, Manuel J. Schaffner * Fachvorgesetzter, Pascal Habegger |
| Themen: | Telefonische Besprechung betreffend Meetings (Expertenbesuch1, Expertenbesuch2, Präsentation IPA) |
| Ergebnis: | Meetings festgelegt auf: 07.05.2014 / 22.05.2014 und 10.06.2014 |

|  |  |
| --- | --- |
| Datum: | 05.05.2014 |
| Typ: | Kurzmeeting |
| Beteiligte Personen: | Fachspezialist Technik, Mathias Leimgruber |
| Themen: | Analyse / Auswahl für Konzepterarbeitung |
| Ergebnis: | Analyse für die verschiedenen Konzeptthemen. Die Entscheidung ist auf folgende Konzeptthemen gefallen:  - Systemanforderung  - Systemarchitektur  - Zugriffskonzept  - Betriebskonzept  - Testkonzept  - Einführungskonzept |

|  |  |
| --- | --- |
| Datum: | 06.05.2014 |
| Typ: | Kurzmeeting |
| Beteiligte Personen: | Fachspezialist Technik, Mathias Leimgruber |
| Themen: | Analyse / Auswahl Kriterien für Variantenentscheid |
| Ergebnis: | Eine Auswahl von verschiedenen Kriterien für den Variantenentscheid wurde getroffen. Bei folgenden Kriterien macht es Sinn, diese für den Variantenentscheid einzusetzen:   * Usability für Entwickler * Usability für weitere Benutzer * Flexibilität * Unabhängigkeit * Aufwand für: * Entwicklung * Einführung * Testing |

|  |  |
| --- | --- |
| Datum: | 06.05.2014 / 07.05.2014 |
| Typ: | Mailing |
| Beteiligte Personen: | Hauptexperte, Manuel J. Schaffner |
| Themen: | Fragen betreffend Anpassung Zeitplan |
| Ergebnis: | Rückmeldung erhalten inwiefern der Zeitplan angepasst werden darf.  Soll-Zeiten werden belassen, hingegen bei den Ist-Zeiten kann es Verschiebungen geben. |

# Abschlussbericht

Wurden die geforderten Ziele erreicht?

## Vergleich Ist/Soll

Ist die Umsetzung wie geplant oder gab es Differenzen?

## Mittelbedarf

Welche Mittel wurden gebraucht oder mussten noch beschafft werden?

## Realisierungsbericht

Gab es Probleme während der Realisation? Ungeplante Sachen zum Vorschein oder Ergänzungen?

## Testbericht

Wie verliefen die Tests (Erfolgreich / weniger Erfolgreich)? Gibt es Fehler, die schwerwiegend sind oder solche die später korrigiert werden können/müssen?

## Fazit zum Projekt

Wie ist das Projekt verlaufen (Objektive) Meinung

Hatte das Projekt Stolpersteine, welche etc…

## Persönliches Fazit

Feedback und Reflexion des Lernenden rund um die IPA.

Was war gut, was weniger? Was habe ich gelernt und was würde ich ev. das nächste Mal anders machen? Ausführlich formulieren.

# Unterschriften Teil 1

Die lernende Person bestätigt mit ihrer Unterschrift diese IPA aus Eigenleistung erbracht und nach den Vorgaben der Prüfungskommission Informatik Kanton Bern erstellt zu haben. Die Angaben im Arbeitsjournal entsprechen dem geleisteten Arbeitsaufwand.

Lernende und Fachvorgesetzte haben das Arbeitsjournal vor der Abgabe zu unterzeichnen und somit dessen Authentizität zu bezeugen. Dies ist Zwingend!!!

|  |  |  |
| --- | --- | --- |
| Datum | Name / OE | Unterschrift |
|  | Michel Weingart |  |
|  | Pascal Habegger |  |

Teil 2: Projektdokumentation

IPA Projektname: Automatische Installation von Demo-Webplattformen

Autor: Michel Weingart

Die Projektdokumentation basiert auf der vereinfachten Projektmethode HERMES 5, welche für die Durchführung von IPAs optimal angepasst wurde. Die Meilensteine wurden im Kapitel 5 definiert.

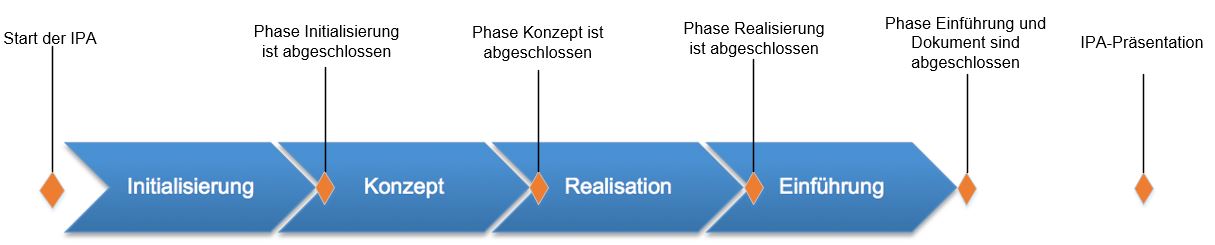


Abbildung : Hermes 5 IPA

# Initialisierung

Die Voranalyse ist ein Klärungsprozess, der mit vertretbarem Aufwand eine Entscheidung über die grundsätzliche Art der Systemrealisierung herbeiführt.

Erstellung und Beurteilung der Situationsanalyse sowie Überprüfung der Zielesetzungen, der Problemstellung und des Untersuchungsbereichs.

Erarbeitung von Lösungsvorschlägen und Abschätzung ihrer voraussichtlichen (Wirtschaft-lichkeit) und Realisierbarkeit. Eine sinnvolle Risikoanalyse für das Projekt, welche Risiken eintreten könnten während -und nach dem Projekt

## Studie Ist Zustand / Soll-Zustand

**Ist:**

Teamraum, Collab, teamraum Intranet und teamraum Web, sind die webbasierten Standardlösungen welche durch 4teamwork entwickelt wurden und immer mit neuen Funktionen erweitert und verbessert werden. Um diesen Prozess optimal umsetzen zu können, sind die neuen Funktionen in einem geordneten Releaseprozess eingebettet.

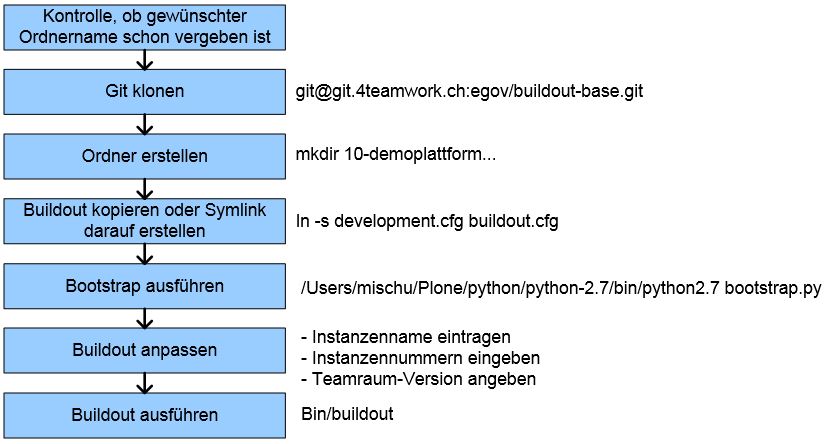
Ein neuer Release wird vom zuständigen Entwickler zwar lokal getestet, doch ist dieser Release für externe Personen nicht zugänglich.

Für das Testen der neuen Releases existiert eine sogenannte Demoplattform, auf welcher sich interessierte Personen registrieren, die neuen Releases testen und Fehler melden können.

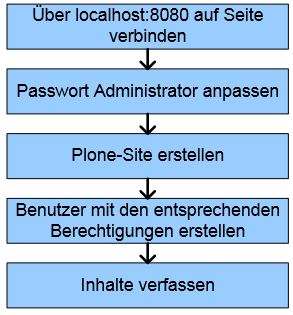
Doch wird diese Plattform weder regelmässig neu aufgesetzt, da der Aufwand doch relativ zeitraubend ist, noch sind Beispielinhalte vordefiniert. Dies kann dazu führen, dass Releases freigegeben werden, ohne dass diese genügend durch „Human Testing“ getestet wurden.

Der aktuelle Prozess um diese Plattform aufzusetzen, sieht folgendermassen aus:

**Durchführungsschritte** **Eingabe Terminal**



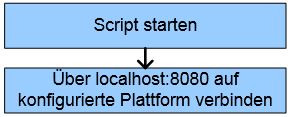
**Weitere Durchführungsschritte**

****

**Soll:**

Eine Demoplattform soll mit den neusten Releases, Beispielinhalten und Beispielbenutzer automatisch aufgesetzt und im Internet bereitgestellt werden können. Dies ermöglicht interessierten Personen, die Lösungen von 4teamwork zu testen und vermindert das Risiko, einen Release freizugeben welcher noch Fehler enthält. Zudem wird die Qualität des Releases erhöht und die Arbeit für den Entwickler erleichtert. Ein weiterer Anwendungsfall, welcher auf demselben Verfahren beruht, ist das Testen von sogenannten Release Candidates.

Der zukünftige Prozess sieht folgendermassen aus:



## Vorgehensziele

Ziel ist es, eine vollfunktionsfähige Lösung zu realisieren, welche eine Demoplattform zu Testzwecken mit Beispielbenutzer und Beispielinhalten bereitstellt.

Die wichtigsten Bearbeitungsschritte um diese Vorgehensziele zu erreichen sind:

|  |  |
| --- | --- |
| Bearbeitungsschritte | Erreicht bis |
| Variantenentscheid | 06.05.2014 |
| Entwicklung Konzept | 19.05.2014 |
| Automatisierung Demoplattform | 20.05.2014 |
| Automatisch werden Benutzer und Inhalte hinzugefügt | 22.05.2014 |
| Testing | 26.05.2014 |
| Freigabe der Software | 27.05.2014 |

## Systemziele

Wie sieht der heutige Zustand aus? Evtl. den Prozess abbilden.

Wie soll das fertige Produkt / die fertige Lösung aussehen? Ev. neuen Prozess abbilden.

Welche Geschäftsprozesse sind durch das Projekt/den Auftrag betroffen?

Detaillierte Auflistung, welche Ziele erreicht werden sollen - abgeleitet aus Projektauftrag.

Das Ziel des fertigen Produkts ist, die Möglichkeit eines einfachen Aufsetzen einer Demoplattform. Ein Entwickler hat die Möglichkeit, ohne Schwierigkeiten und praktisch ohne Zeitaufwand, die Plattform mit dem neusten Release zur Verfügung zu stellen. Dies ermöglicht eine Qualitätsverbesserung der Releases und einen Zeitgewinn für die Entwickler. In der untenstehenden Tabelle sind wichtigsten Ziele erläutert.

|  |  |
| --- | --- |
| Ziele | Erreicht bis |
| Benutzer sollen automatisch erfasst sein | 21.05.2014 |
| Inhalt soll automatisch erfasst sein | 21.05.2014 |
| Automatisches und einfaches aufsetzen der Demoplattform | 26.05.2014 |
| Qualitätsgewinn | Nach Projektabschluss |
| Aufwand für Entwickler wird vermindert | Nach Projektabschluss |

## Anwendungsbereich

Die Lösung wird von Entwicklern von 4teamwork genutzt um einen Release auf einer Demoplattform aufzusetzen. Der Anwendungsbereich könnte eventuell auch auf definierte Externe Benutzer erweitert werden.

## Anforderungen

Unterschiede: Ziele/ Anforderung

􀀂 Ziel:

􀀁 Ein zu erreichender Zustand

􀀁 Meist in umfassendem Sinn gemeint

(Was wollen wir bis wan?)

􀀂 Anforderung:

􀀁 Eine zu erreichende Eigenschaft

􀀁 Meist konkret

􀀁 Beschreibt in der Regel eine für die Erreichung eines Ziels

notwendige Bedingung oder Eigenschaft

(Wie machen wir es)

### Funktionale Anforderungen

Funktionale Anforderungen beschreiben gewünschte Funktionalitäten (was soll das System tun/können) eines Systems bzw. Produkts, dessen Daten oder Verhalten.

Folgende funktionalen Anforderungen werden an das fertige Produkt gestellt:

* Eine Plattform wird automatisch aufgesetzt
* Die Plattform enthält nach Initialisierung Testbenutzer
* Die Plattform enthält nach Initialisierung Testdaten
* Die Plattform ist nach Initialisierung bereit zum Testen

### Nicht funktionale Anforderungen

Nichtfunktionale Anforderungen sind Anforderungen, an die "Qualität" in welcher die geforderte Funktionalität zu erbringen ist.  
  
Qualität im vorgenannten Sinn meint beispielsweise

wie die Funktionalität ausgeführt werden soll (z.B. Reaktionszeit)

Bedingungen unter denen die Funktionalität ausgeführt wird (z.B. 7x24 Std.)

Oder einen schnell zu bedienen GUIs (z. B Software Ergonomie)

Folgende Nicht funktionalen Anforderungen werden an das fertige Produkt gestellt:

* Eine einfache Bedienung ist gewährleistet
* Die Installation läuft zuverlässig / Stabil / Fehlerfrei auf der geforderten Umgebung
* Die Lösung kann ohne grossen Aufwand erweitert werden
* Die Lösung kann ohne grossen Aufwand portiert werden

### Wirtschaftlichkeit

**Nutzen:** Die Qualität der Releases wird gesteigert und die Arbeitszeit, welche benötigt wird eine Demoplattform aufzusetzen und diesen Release testen zu können, erheblich reduziert.

**Kosten:** Die Kosten belaufen sich auf 5 Arbeitstage. Eventuell werden weitere Kosten anfallen um die Lauffähigkeit des Skriptes auf einem Server zu gewährleisten.

## Risikoanalyse

Welche Risiken ergeben sich, wenn das Projekt nicht realisiert wird? Was ist, wenn das Projekt scheitert? Welches sind die grössten Risiken bei diesem Projekt.

Die Demoplattformen können jederzeit auch manuell durch die Entwickler, mit den neusten Releases, aufgesetzt werden. Deshalb besteht kein Risiko falls dieses Projekt nicht durchgeführt werden kann.

## Varianten

Wichtig ist: Mögliche Varianten vorzustellen und sich für eine Variante zu entscheiden..

Es bestehen mehrere Möglichkeiten die gewünschte Demoplattform bereitzustellen. Entweder wird diese auf einer Virtuellen Maschine installiert und das Image bei jedem Release neu konfigurieren, oder es wird ein Script erarbeitet, welches ermöglicht, die Demoplattform ohne Aufwand aufzusetzen. Anhand folgender Beurteilungskriterien, wird eine von den zwei Varianten ausgewählt:

* **Usability:** Wie bedienerfreundliche ist das Aufsetzten der Demoplattform für die Entwickler? Kann die Demoplattform auch von weiteren Personen benutzt werden?
* **Flexibilität:** Welcher Aufwand wird benötigt, um die Variante zu erweitern oder anzupassen?
* **Unabhängigkeit:** Kann das Aufsetzten der Demoplattform unter verschiedenen Betriebssystemen durchgeführt werden?
* **Entwicklungsaufwand:** Welcher Aufwand wird benötigt um die Variante zu entwickeln?

### Variante 1

Auf einer virtuellen Maschine wird ein vorbereiteter Release einer Webanwendung, inklusiv Beispielbenutzer und Beispieldaten, installiert. Diese virtuelle Maschine kann verteilt und zwecks Tests zur Verfügung gestellt werden. Das Image kann regelmässig mit einem neuen Release neu geladen werden. Folgende Vorteile und Nachteile konnte ich zu dieser Variante herausfiltern:

**+ Aufwand:** Der Aufwand um eine Virtuelle Maschine mit Unix-basiertem Betriebssystem aufzusetzen und ein Script zur Automatisierung zu erstellen ist relativ gering.

**+ Usability für weitere Benutzer:** Eine externer Benutzer kann ein Image einbinden und die Plattform nutzen.

**+ Unabhängigkeit:** Ein Image kann auf verschiedensten Betriebssystemen aufgesetzt werden.

**- Flexibilität:** Ein Image ist meiner Meinung nach recht starr und eine Erweiterung recht aufwändig. Um einen neuen Release zu erfassen, müsst das Image jedes Mal angepasst werden.

**- Usability für Entwickler:** Recht umständlich, da die Entwickler bei 4teamwork nicht mit Images arbeiten und dies ein Mehraufwand bedeuten würde.

### Variante 2

Auf einer Linux-Umgebung, werden per Skripts die Webanwendung automatisch installiert und Beispielbenutzer und Beispielinhalte erfasst. Folgende Vorteile und Nachteile konnte ich zu dieser Variante herausfiltern:

**+ Usability Entwickler:** Die Ausführung des Skriptes bedeutet für den Entwickler keinen Aufwand, da dies auf einer bekannten Umgebung ausgeführt wird.

**+ Aufwand:** Der Aufwand um ein Script zu erstellen ist gering.

**+ Unabhängigkeit:** Ein Python-Script kann nur auf Unix-basierten Betriebssystemen ausgeführt werden.

**- Flexibilität:** Ein Script kann schnell angepasst und erweitert werden. Eine neue Version einzutragen wird von einem Entwickler in kürzester Zeit durchgeführt.

**- Usability für weitere Benutzer:** Ein externer Benutzer kann das Skript ohne Vorkenntnisse wahrscheinlich nicht ausführen.

## Variantenentscheid



Tabelle 2: Variantenentscheid

### Begründung

Aufgrund des Variantenentscheids, habe ich mich zur Umsetzung per Variante 2 entschieden. Ein sehr grosser Vorteil der Variante per Script ist die Kombination der Flexibilität und Usability für die Entwickler. Das Script kann Eingabeparameter entgegennehmen, welche für die Variante per Virtueller Maschine nicht definiert werden könnten.

## Informationssicherheit und Datenschutz (ISDS)

Es besteht keine Gefährdung von Daten und Systemen. Das Script wird lokal ausgeführt und kann keine Beschädigung des Systems erwirken. In einer produktiven Umgebung (Server) muss sich jeder Benutzer authentifizieren was die Sicherheit gewährleistet.

## Lösungen suchen

Die Lösungssuche ist ein strukturierter und kreativer Prozess, in welchem nach möglichen Ansätzen gesucht wird, welche die Umsetzung der Systemziele unterstützen.

Der Entscheid über den Lösungsvorschlag schliesst die Lösungssuche ab.

Um Lösungsansätze zu finden wie das Script funktionieren / aussehen und aufgebaut sein soll, wurde die Kreativitätstechnik „Brainstorming“ gewählt. Für 5 Minuten wurden alle Ideen zu folgender Fragestellung gesammelt:

*Was muss das ideale Script beinhalten und wie funktioniert dieses ?*

Die wichtigsten Kriterien wurden herausgefiltert und werden für den Lösungsentscheid verwendet.



**Entscheid über den Lösungsvorschlag**

Das Script wird folgende Elemente beinhalten:

* Automation muss gewährleistet sein, eventuell mit Parameterübergabe
* Die Benutzererstellung muss gewährleistet sein
* Die Inhaltserstellung muss gewährleistet sein
* Berechtigung des Scripts muss gewährleistet sein
* Das Script muss von jedem Standort auf dem Server / lokal ausführbar sein
* Ausführung des Scripts überschreibt frühere Demoplattform
* Das Script soll wird über das Terminal angestossen
* Kommentare müssen im Code eingefügt werden
* Das Script soll erweiterbar sein
* Eventuell Logfile / Protokoll
* Code soll ausgelagert werden falls möglich
* Im Script kann eventuell weiteres Script angestossen werden
* Wichtige Punkte sollen im Script mit Markierungen aufgezeigt werden

## Freigabe Konzept

Die Phase Konzept wird freigegeben:

|  |  |  |
| --- | --- | --- |
| Datum | Name / Auftraggeber | Unterschrift |
| 06.05.2014 | Pascal Habegger |  |

# Konzept

In der Konzepterarbeitung werden die Grundlagen für die Realisierung und Einführung eines Informatiksystems entwickelt.

Das Konzept wird schrittweise mit folgenden Schritten entwickelt.

Es ist wichtig, die Ergebnisse so weit zu deklarieren, dass damit die Systemarchitektur bestimmt werden kann. Abgestimmt mit der schrittweisen Entwicklung des Konzepts werden die Fertigprodukte? Sachmittel evaluiert.

* Systemanforderungen
* Systemarchitektur
* Materialbeschaffung
* (Systemintegrationsplan)
* Einführungskonzept
* (Datenmigration)
* Ausbildungskonzept
* (Wirtschaftlichkeit)
* ISDS-Konzept
* Backup-Konzept
* Test-Konzept
* Etc..

Achtung nicht alle Konzepte müssen gemacht werden, dies gilt rein als Idee…

Wichtig ist: Was steht im Aufgabenbeschrieb bzw. was wird von Pkorg verlangt siehe Kriterienkatalog. (Achtung Meilensteine die von der Projektmethode angegeben sind, müssen erreicht werden).

In dieser Phase steht die Erarbeitung der Grundlagen betreffend der Realisierung und Einführung des Projektes im Vordergrund.

## Zusammenfassung

xxxxxxx

## Situationsanalyse

Der Sachverhalt wird durchleuchtet, die Ausgangslage erläutert und verschiedene Aspekte zur Situation aufgezeigt. Die Situationsanalyse stellt die Grundlage für die Suche nach einer Lösung dar.

### Ausgangslage

Um für externe Benutzer neue Releases zu testen, steht eine Demoplattform zur Verfügung. Diese wird von Entwicklern zu unregelmässig neu aufgesetzt was dazu führen kann, dass ein Release freigegeben wird, welcher vermeidbare Fehler enthält.

### Ziele und Wünsche

Das Ziel des Projekts ist, die automatische Bereitstellung von Demoplattformen, auf welchen neue Releases getestet werden können. Benutzer und Inhalte werden zudem automatisch eingepflegt. Zudem wäre es wünschenswert, dass die Demoplattform auf weiteren Servern eingepflegt werden kann.

### Zielgruppen

Die Zielgruppe ist in erster Linie externe Benutzer, welche sich auf der Demoplattform anmelden und die Releases testen können, sowie die Entwickler welche praktisch ohne Aufwand, eine Demoplattform mit den neusten Releases initialisieren können.

### Angebot

Das Angebot besteht darin, ein Script zu erstellen, welches die Automation gewährleistet und die Demoplattform mit Beispielinhalt und Beispielbenutzer aufsetzt.

### Zeitrahmen / Budget

Der Zeitrahmen umfasst 5 Arbeitstage, das Projekt wird am 27.05.2014 abgeschlossen.

### Übergeordnete Ziele

Neuste Releases werden zum Testen zur Verfügung gestellt, was dazu führen soll, dass Fehler in den Releases besser erkannt werden. Somit sollen die Qualität der Releases und auch die Qualität der Produkte von 4teamwork verbessert werden.

### Integration in die bestehende Umgebung

Das Script wird als solches auf einem zur Verfügung gestellten Server implementiert. Dazu werden wahrscheinlich noch kleinere Anpassungen am Quellcode nötig sein. Die Ausführung des Scripts erfolgt, nach der Integration, über die Kommandozeile.

### Chancen

Benutzer erwarten eine konsistente und Fehlerfreie Applikation. Die Durchführung des Projekts ermöglicht eine weitere Annäherung an diese Erwartungshaltung. Neue Releases werden dank „Human Testing“ verbessert und Fehler erkannt und ausgemerzt. Dies trägt einen Anteil dazu bei, die Produkte verbessert freizugeben und deren hohe Qualität zu gewährleisten.

Der Markt umfasst teilweise ähnliche Lösungen wie diejenigen, welche von 4teamwork angeboten werden. Wenn die Qualität der Produkte gewährleistet werden kann, ist die Konkurrenzfähigkeit garantiert.

### Zusammenfassung

Das Projekt welches bis zum 27.05.2014 abgeschlossen wird, ermöglicht eine Verbesserung der Produkte von 4teamwork in verschiedenen Bereichen.

* Entwickler benötigen weniger Aufwand um eine Demoplattform zur Verfügung zu stellen. Die gewonnene Zeit kann wiederum in die Produkte von 4teamwork einfliessen, was die Markt- und Konkurrenzfähigkeit steigert.
* Die Releases werden mit weniger Fehler veröffentlicht was wiederum die Markt- und Konkurrenzfähigkeit steigert.

## Allgemeine Anforderungen

Um die Lauffähigkeit des Scripts gewährleisten zu können, müssen gewisse Anforderungen erfüllt sein.

### Funktionale Anforderung

* Ein Python muss installiert sein, welches mit Plone umgehen kann
* Dem Script muss der Pfad zum python bekannt sein
* Das Script muss die entsprechenden Berechtigungen besitzen um eine Interaktion durchführen zu können
* Das Script muss sich auf das entsprechende Git Repository authentifizieren können

### Nicht funktionale Anforderungen

* Das Script soll Eingabeparameter entgegennehmen können
* Das Script soll Fehler dokumentieren

## Systemarchitektur

### Physikalische Hardware / Software Lokal

|  |  |
| --- | --- |
| Beschreibung | Daten |
| Typ | MacBook pro 5.4 |
| Betriebssystem | OS Version 10.9.2 |
| Motherboard | Keine Angabe |
| CPU | 2.53 Ghz Intel Core Duo |
| Festplattentyp / Festplattenspeicher | * SSD 240Gb * HDD 250Gb |

Falls die Implementation bis zum Ende dieses Projekts auch Serverseitig durchgeführt werden kann, wird auch die Systemarchitektur des betroffenen Servers benötigt.

### Physikalische Hardware / Software Server

|  |  |
| --- | --- |
| Beschreibung | Daten |
| Typ | Server |
| Hostname | poseidon.4teamwork.ch |
| Interne IP-Adresse | 212.103.94.18 |
| Motherboard | IBM |
| Betriebssystem | CentOS 5.10 |
| CPU | 2.80GHz Intel(R) Xeon(R) |
| Festplattenspeicher disk1 | IBM 175GB |
| Festplattenspeicher disk2 | 145GB |

## Zugriffskonzept

Bei der Erarbeitung des Zugriffkonzepts geht es darum, einerseits die grundsätzlichen Berechtigungen der verschiedenen automatisch erzeugten Benutzer auf der Demoplattform abzubilden und anderseits die benötigten Berechtigungen des Scripts darzustellen. Um ersteres verständlich darstellen zu können, wird zuerst die Datenstruktur der Demoplattform und anschliessend die Berechtigungen der Benutzer vereinfacht aufgezeigt.

### Datenstruktur

**Hier kommt die Struktur**

**Erläuterung der Typen betreffend Struktur**

|  |  |  |
| --- | --- | --- |
| Name | Typ | Funktion |
| Intranet | PloneSite |  |
| Startseite | Spezielle Inhaltsseite |  |
| Arbeitsräume | Ordner |  |
| News | Inhaltsseite |  |
| Kontakte | Ordner |  |
| Organisation | Inhaltsseite |  |
| Subsite Teaserportlet | KeineTypenbezeichnung |  |
| Newsportlet | KeineTypenbezeichnung |  |
| Merkzettel | KeineTypenbezeichnung |  |
| Kalender | KeineTypenbezeichnung |  |
| Dynamische Portlets | KeineTypenbezeichnung | Dynamische Inhalte (Z.B News) werden in einem dynamischen Portlet dargestellt. |
| Statische Portlets | KeineTypenbezeichnung | Statische Inhalte, meist zur Information werden in einem statischen Portlet dargestellt. |
| Interne News | Newsordner | News werden erfasst und können in einem dynamischen Portlet dargestellt werden. |
| Maria Meier | Kontakt | Wird zum Erfassen eines Kontakts verwendet. |
| Inhaltsseite |  | Wird zur Strukturierung der Plattform verwendet. |
| Adresenblock | KeineTypenbezeichnung | Ein Inhaltselement innerhalb einer Inhaltsseite |
| Auflistungsblock | KeineTypenbezeichnung | Ein Inhaltselement innerhalb einer Inhaltsseite |
| Bild | KeineTypenbezeichnung | Ein Inhaltselement innerhalb einer Inhaltsseite |
| Datei | KeineTypenbezeichnung | Ein Inhaltselement innerhalb einer Inhaltsseite |
| Textblok | KeineTypenbezeichnung | Ein Inhaltselement innerhalb einer Inhaltsseite |

### Berechtigungen

Die Rechte auf der Demoplattform werden anhand eines definierten Workflows vergeben. Es wird bei den Berechtigungen zwischen globalen und lokalen Berechtigungen unterschieden. **Globale Berechtigungen** sind nicht Typenspezifisch (beziehen sich nicht auf Inhaltstype) und werden an untergeordnete Elemente vererbt. Da die globalen Berechtigungen auf der obersten Ebene vergeben werden, ist die Vererbung auf der ganzen Plattform vergeben. Eine globale Berechtigung kann nicht unterbrochen werden.

**Lokale Berechtigunen** sind typenspezifisch(beziehen sich auf Inhaltstype) und können an untergeordnete Elemente vererbt werden. Diese Vererbung kann unterbrochen werden.

Da für die Demoplattform nur ein Benutzer pro Rolle erstellt wird und somit keine individuellen Berechtigungen auf Elemente vergeben werden müssen, werden globale Berechtigungen wie folgt vergeben:

|  |  |  |  |
| --- | --- | --- | --- |
| Benutzer | Rolle | Berechtigung | Funktion |
| Administrator | Administrator | Global | Administriert die Inhalte der Plattform. Er kann die Struktur anpassen und Module hinzufügen. |
| Mitarbeiter\_Intern | User | Global | Kann Inhalte erfassen / bearbeiten und löschen |
| Mitarbeiter\_Extern | Reader | Global | Hat Leserechte auf Inhalte |

## Betriebskonzept

Im Betriebskonzept wird beschrieben, wie das Script effektiv betrieben werden soll.

### Erstellen und Einpflegen des Scripts

Das Script wird einem Branch des Pakets „eGov buildout“ erstellt. Es wird ein Ordner „script“ hinzugefügt, welcher das Script „teamraum\_demo.py“ enthält.

Zudem wird ein Branch des Pakets „eGov core Modul“ mit dem Profil „teamraum\_demo“ erstellt und der Inhalt in diesem Branch definiert.

Ev. Visio vom Prozess eines Pakets.

### Qualitätssicherung

Die Qualitätssicherung wird mit der Versionsverwaltung von Git gewährleistet. Ein Branch wird in der Regel von einer weiteren Person kontrolliert und von dieser in den Master gemerged.

### Sicherung / Archivierung des Scripts

Die Sicherung und Archivierung wird über das Git Repository, welches die Pakete „eGov core Modul“ und „eGov buildout“ enthält, gewährleistet. Alle früheren Versionen werden aufgelistet und ein Backup des Scripts ist jederzeit gewährleistet.

### Versionierung

Nachdem das Script die geforderten Funktionen erfüllt, wird die Version 1.0 in den Master des betroffenen Pakets gemerged. Für eine neue Version wird wiederum ein Branch des Masters erstellt und dieser nach Fertigstellung gemerged.

### Hotfixes

Hotfixes werden ähnlich wie die Implementierung neuer Versionen gehandelt. Wird ein Fehler im Script entdeckt, wird ein Branch des Masters erstellt, in welchem der Fehler behoben wird. Dieser wir nach Fertigstellung wieder in den Master gemerged.

### Fehlerhandling

Bei einem Fehler wird das Script abgebrochen, damit dieses keine Prozesse unnötig blockiert. Ein weiteres Ausführen des Scripts werden die schon erstellten Inhalte überschrieben.

### Betrieb auf Client

Das Script wird so aufgebaut, dass dieses lokal auf einem MacBook pro funktionieren wird.

### Betrieb auf dem Server

Die Ausführung des Scripts auf dem Server funktioniert ähnlich wie die Ausführung auf einer lokalen Maschine. Einige kleinere Anpassungen sind wahrscheinlich nötig um die Funktionalität gewährleisten zu können. Die Implementation auf einem Server ist nicht Bestandteil meiner IPA, doch falls es zeitlich noch möglich ist wird diese in einem Release 2.0 des Scripts, während der Realisierung durchgeführt.

## Systemdesign erstellen

Das Systemdesign beschreibt und dokumentiert den Ausbau und die Funktionsweise des Informatiksystems:

* Systemanforderungen
* Systemarchitektur
* Systemintegrationsplan
* Migrationsdesign

## Schutzmassnahmen erarbeiten

ISDS-Konzept

## Testkonzept.

Das Testkonzept definiert die Testmethoden, die Testziele, den Testrahmen und die Testvorgehen, und dient der effizienten Planung und Durchführung der einzelnen Tests.

Das Testkonzept enthält alle Rahmenbedingungen und Verfahrensbeschreibungen zur Durchführung der Tests:

* Testziele
* Testrahmen
* Testvorgehen
* Testmethoden und Testfälle
* Was passiert bei einen Fehler (Re-Testing)?

Die Testprozedur ist eine Arbeitsanleitung, die exakte Anweisung für Tests (Testspezifikation) und wird für jeden einzelnen Test erstellt:

Testobjekt und Testfall

Arbeitsanleitung

Vorbereitung

Voraussetzung

Konfiguration

Durchführung

Resultatsicherung und -auswertung

(Wichtig Tests sind individuell, beachtet was wird von den Kriterien verlangt.)

BSP: Java Entwickler macht z. B ein Juntis-Tests Ein System Administrator ev. ein Sicherheitstest. Es müssen in der Regel mehrere verschieden Tests durchgeführt werden!

# Realisierung

Die Phase Realisierung dient zur Erstellung des Systems und schafft die Voraussetzungen für die nachfolgende Einführung

## System erstellen

Das Verfahren zur Systemerstellung wird nochmals überprüft und allenfalls angepasst:

* System entwickeln
* Ausbildungen planen und Schulungsmaterial spezifizieren
* Anwendungshandbuch und Schulungsmaterial erstellen
* Betriebshandbuch erstellen

## Testprotokoll

* Die Testprotokolle fassen die Ergebnisse zusammen:
* Testobjekt
* Angaben zur Durchführung (konnten alle Tests vom Konzept durchlaufen werden?
* Testresultate
* Was passiert mit einem „Failtesting“? „Re-Testing“ oder leichter Fehler?
* Testfazit und Empfehlung

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nr. | Testobjekt | Testbeschreibung | Erwartung | Ergebnis | Visum |
| 1B |  |  |  |  |  |
| 2W |  |  |  |  |  |

Tabelle 3: Testprotokoll

## Einführung vorbereiten

Die betrieblichen und organisatorischen Änderungen, welche bei der Einführung auftreten, müssen vorbereitet werden:

* Sicherstellen der Produktionsumgebung
* Datenmigration
* Ausbildung planen
* Ausbildungsmaterial und Handbücher erstellen

## Schutzmassnahmen umsetzen

Die vorgesehenen Massnahmen zum Schutz des Systems werden umgesetzt, deren Umsetzung überprüft und falls deren Wirkung durch gezielte Verbesserungen erhöht, so dass die Anforderungen an die Sicherheit und den Datenschutz erfüllt sind.

ISDS-Konzept überprüfen und ergänzen

Schutzmassnahmen durchführen

Schutzmassnahmen verbessern

# Einführung

Die Phase Einführung dient zur Installation des Informatiksystems und zur Aufnahme des Betriebs sowie Abschluss des Projekts (Bericht Teil 1).

## System einführen

* Informatiksystem installieren und verteilen
* Benutzer ausbilden
* Informatiksystem aktivieren
* Prozesse und Organisation aktivieren

# Quellenverzeichnis

Hinweis: An dieser Stelle muss ein Literatur- und Quellenverzeichnis eingefügt werden.

(es kann auch mit der Fussnote ein Hinweis auf die Quelle gemacht werden, diese muss aber im Quellen VZ ersichtlich sein

Internet Quelle:

Name des Autors (falls erkennbar), „Titel der Seite“, Webadresse, Datum des letzten Zugriffs

Beispiel Internet

Quellenangabe „Koala“, http://de.wikipedia.org/wiki/Koala, 22.03.2008

Buch Quelle:

Erklärung Name des Autors: Titel. Verlag, Jahr, Seite, auf der der zitierte Text steht.

Beispiel Buch

Quellenangabe Ernst Walter Bauer: Humanbiologie. Cornelsen, 2006, S. 50.

# Glossar

Alphabetisch sortiert

|  |  |
| --- | --- |
| Begriff | Bedeutung |
|  |  |
|  |  |
|  |  |

Tipp: Word 2010/13 Layout/Sortieren nach… ;-)

# Unterschriften für Abnahme

Lernende und Fachvorgesetzte haben die Dokumentation vor der Abgabe zu unterzeichnen und somit die Richtigkeit zu bezeugen. Der Fv hat die IPA abgenommen, und bewertet diese anschliessend.

|  |  |  |
| --- | --- | --- |
| Datum | Name / OE | Unterschrift |
|  |  |  |
|  |  |  |

Achtung:

Auf dem Deckblatt Unterschrift des FV und des Lernenden

(Blaues den Experten aushändigen, nicht unterschreiben, und Gelbes FV, unterschrieben)!

# Anhang

Ausgedruckter Code, Sitzungsprotokolle, usw.: Vollständig aufführen und der Dokumentation beilegen.

Listings von Scripten und Programmen. Die Eigenleistung der Kandidatin oder des Kandidaten muss vollständig dokumentiert sein. Automatisch generierten Code weglassen, wenn für das Verständnis nicht zwingend nötig.

Falls Handbücher erstellt wurden, können diese hier als Anhang beigelegt werden.

Firmenstandards zwingend in den Anhang!

* Die Bearbeitungsschritte der Lösung werden geloggt
* Die Lösung läuft auf einem Server
* Die Lösung kann für definierte externe Benutzer verwendet werden